National Science Digital Library Conference
Chicago November 15, 2004

Getting a Leg Up on the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)

Chair:

Muriel Foulonneau, visiting project coordinator for the CIC metadata portal, University of Illinois at Urbana-Champaign, mfoulonn@uiuc.edu

Presenters:

Naomi Dushay, National Science Digital Library project, Cornell University, Naomi@cs.cornell.edu

Edward Almasy, co-director, Internet Scout Project, University of Wisconsin – Madison, ealmasy@scout.wisc.edu

Thomas Habing, University of Illinois at Urbana-Champaign, research programmer thabing@uiuc.edu

Lyle Barbato, Compadre project, database administrator, American Association of Physics Teachers, lbarbato@aapt.org

Handout written with the contributions of presenters and **Diane Hillmann**, Director of Library Services and Operations, National Science Digital Library project, Cornell University

The NSDL Fact Sheet

"*[t]he Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content*".(OAI FAQ)

The OAI protocol allows a data provider (you) to share metadata through an OAI repository with a service provider (such as NSDL). The service provider launches a program called a harvester to collect your data regularly and present it in a new application such as the NSDL portal. The OAI protocol is based on 3 standards: HTTP, XML and the Dublin Core Metadata Element set.

Learn about OAI

The **OAI protocol:**
http://www.openarchives.org/OAI/openarchivesprotocol.html
Guidelines to OAI implementers:
http://www.openarchives.org/OAI/2.0/guidelines.htm
An **OAI Tutorial:** http://www.oaforum.org/tutorial/
The **NSDL OAI FAQ**:
http://metamanagement.comm.nsdlib.org/NSDL_OAI_FAQ.html

The **DLF OAI and Metadata Sharing Best Practices** documents (as of 10/2004, very much a work in progress):
http://oai-best.comm.nsdl.org/cgi-bin/wiki.pl?TableOfContents

Learn about XML

NSDL XML FAQ at
http://metamanagement.comm.nsdlib.org/NSDL_XML_FAQ.html
A brief overview of XML, Tim Cole and Thomas Habings, UIUC
http://dli.grainger.uiuc.edu/Publications/TWCole/AALL_2000/PDF/Overview.pdf
XML tutorial, Tim Cole and Thomas Habings, UIUC,
http://dli.grainger.uiuc.edu/cdp/ColeHabingXMLTut.ppt

You will need to know how to send an XML response over HTTP and the basics of GET and POST requests.

Several examples of OAI services

- The **National Science Digital Library** displays 372 scientific collections of online resources for science, technology, engineering and mathematics education http://www.nsdl.org/
- **OAIster** allows searching over 3.5 million on-line resources on all subjects from over 350 institutions http://oaister.umdl.umich.edu/o/oaister/
- **American South**, a scholarly discovery service for research materials related to the cultures and histories of the American South http://www.americansouth.org/

- **The UIUC Digital Gateway to Cultural Heritage Materials** http://nergal.grainger.uiuc.edu/cgi/b/bib/bib-idx
- The **CIC metadata portal** that allows multiple representations of Midwestern university resources http://cicharvest.grainger.uiuc.edu/

Several NSDL data providers

- **ComPADRE** : resources for physics and astronomy education http://www.compadre.org/portal/index.cfm
- **Internet Scout** http://scout.wisc.edu/
- **The Pacific resources for education and learning** http://www.prel.org/
- **K-12 Science** http://www.sciquest.com/k12/
- **DLESE** Digital Library for Earth System Education http://www.dlese.org/dds/index.jsp
- **American Natural Science in the First Half of the Nineteenth Century** (Academy of Natural Sciences of Philadelphia) – static repository implementation http://www.acnatsci.org/library/collections/imls/index.html

OAI Static repositories

"The OAI Static Repository and OAI Static Repository Gateway [provide] a low barrier solution for Data Providers to make metadata collections available to the world. The Data Provider writes an XML file with a specific format - an OAI Static Repository - which is made OAI-PMH harvestable through intermediation of software - an OAI Static Repository Gateway - operated by a third party." [Patrick Hochstenbach and Herbert van de Sompel]

- **Specification for an OAI Static Repository** http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm
- **How to Run your own OAI Static repository and repository gateway** *by Patrick Hochstenbach and Herbert van de Sompel* http://srepod.sourceforge.net/

To build a static repository, you need :

1- a **valid XML file** (updated as necessary) available via HTTP, including all the metadata you need to expose
2- an **XML Schema** for any custom metadata formats
3- to **register** at a static repository gateway

Looking for OAI support?

The NSDL all projects mailing list http://comm.nsdl.org/mailman/listinfo/nsdl-all-projects
The OAI-implementers mailing list
http://www.openarchives.org/mailman/listinfo/OAI-implementers
The NSDL Metadata Repository ingest team mr-ingest@nsdl.org

Good and Best practices for OAI Data Providers

Ten Commandments for Metadata Quality

The quality of a service provider's search portal (or other service offered) is dependent on the consistency, quality and richness of the metadata you provide, so first and foremost, *think of ways you can help service providers highlight the wonderful treasures you hold*. Other recommendations to consider:

1. Provide richer metadata formats in addition to Simple Dublin Core!

For instance, NSDL has its own Qualified Dublin Core schema (nsdl_dc). Or you may use other standard "native" metadata formats for which an XML Schema exists (see: http://metamanagement.comm.nsdlib.org/contributing.html#standards for more information).

2. Use and specify controlled vocabularies in your metadata

Examples are LCSH, the NASA thesaurus or a controlled vocabulary for audiences, such as the NSDL's Education Level (see http://ns.nsdl.org/schemas/ed_type/ed_type_v1.00.xsd). Using and specifying controlled vocabularies helps the service provider understand and use your content effectively. Note that controlled vocabulary terms can be used in Simple DC, but to specify the vocabulary used, you must provide Qualified DC or a similarly rich metadata format.

3. Make sure your metadata refers explicitly to the resource it describes

If your dc:identifier fields do not include an identifier (preferably a URI) that can link directly to the resource, your metadata may not be useful to the NSDL. Internal database identifiers are useless outside your context. Remember, your OAI identifier refers to your metadata record, but the dc: identifier refers to the resource you're describing with the metadata.

4. Don't embed additional labels within Simple Dublin Core elements

Some data providers who find Simple DC insufficient and richer formats too daunting sometimes embed labels (usually Qualified DC element refinements) at the head of their value strings. This practice has been deprecated by DCMI, and is impossible for NSDL (as an aggregator relying on automated processing) to either detect, use, or eliminate. [Note: If you're tempted to do this, you should definitely be using a richer format in addition to Simple Dublin Core.]

5. Be consistent in form of personal and corporate names

Make sure that the form of names is consistent (either surname, forename or direct order). Consistent use of name forms will allow future parsing of those names for improved precision and recall. Remember that affiliations and email addresses are not names, and should not be included with personal names.

6. Limit or eliminate the use of HTML within your metadata

HTML tagging within metadata behaves unpredictably when reused in contexts not optimized for HTML presentation. Also keep in mind that some HTML tagging conventions may not be valid within XML, and might cause your data to fail validation routines.

7. Consider using separate element instances for multiple values

In general, separate instances are easier to parse and interpret, and machines don't care how long your record is. On exception to this is keywords in dc:subject, where multiple values are a reasonable choice. Note that both DCMI and NSDL recommend using semi-colons, not commas, to separate values when in multiples within a single element instance. This ensures that there is no ambiguity when commas are used for other purposes within repeated values, for instance to indicate surnames and forenames.

8. Eliminate empty elements, "content-free" elements and useless characters before exposing your metadata

Metadata contributed to NSDL contains various defaulted strings signifying "no information available" -- generally something like *unknown* (sometimes abbreviated or misspelled), or values comprised solely of stray characters such as dashes or hyphens. Sometimes these result from crosswalking, defaults when applications require certain elements to be filled, or just misunderstanding about options. Be aware that these no-value values degrade the user experience, and make sure you're not distributing these in your metadata.

9. Ensure continuous metadata management and quality control

Include support for metadata creation and maintenance in initial project planning rather than as an afterthought. Quality control should emphasize consistency and clarity to enable reuse of metadata in other contexts.

10. Make sure your metadata makes sense outside your context

Remember that if your metadata is aggregated by others, the context that your collection may provide may be lost. Make sure that topics, associations with institutions and people, and formats are not *assumed*, but reflected specifically in your metadata records. If you're new to metadata and are not sure whether you're doing the right thing, ask for help. DCMI has an AskDCMI service (http://askdcmi.askvrd.org), and NSDL repository staff are happy to review your metadata if you ask (mailto: mr-ingest@comm.nsdl.org)

Principles to implement an OAI repository

For the NSDL to successfully harvest your metadata, your OAI Repository must be compliant with the OAI-PMH version 2.0 specification.

How you organize content within your repository, the completeness and validity of your repository's responses and how well you monitor and update your system directly impacts how well service providers can represent reliable, up-to-date and complete collections to users.

1. Each metadata record must pass XML validation

XML encoding MUST be valid and the record MUST comply with a valid XML Schema
http://metamanagement.comm.nsdlib.org/NSDL_XML_FAQ.html,
http://metamanagement.comm.nsdlib.org/NSDL_Encoding_FAQ.html
http://www.oaforum.org/tutorial/ (this provides a very good XML section)

2. Choose appropriate OAI identifiers

Identifiers should uniquely identify your record, be of a reasonable length (< 128 bytes) and be a valid URI string. We strongly recommend using the OAI Identifier Format (http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm)

3. Support persistent OAI deleted records and second-based datestamps

If you simply remove an item from your OAI repository, harvesters that grab updates ("what has changed since I last harvested?") will never realize the item has been deleted from your repository. Without indicating deleted records, the harvester must periodically reharvest your entire repository to avoid perpetuating records you have deleted. Persisting deleted records allows incremental harvesting to get this information, saving time and bandwidth for everyone.

4. Define sets according to your service provider's need and describe them clearly

If the NSDL is only interested in a portion of your content, please make a distinct harvestable set(s) for that portion. Also, consider creating an OAI set for each distinct collection you hold. Do not rely on the set membership for the only topical context for your data: set membership does not substitute for subject terms in the metadata records.

5. Use resumption tokens appropriately

Resumption tokens allow metadata record harvesting to avoid huge HTTP responses; instead, large lists are broken into chunks. They avoid server timeout when the number of records sent for the OAI request is too large, as well as providing an easy way to recover from a temporary internet glitch. If responses are too big, they may be difficult to retrieve reliably via HTTP. If responses are too small, a great deal of extra network traffic is required to harvest your records. A reasonable size for a single HTTP response is between 500,000 and 2,500,000 bytes – you will have to determine the number of records based on your typical record size.

6. Validate and register your repository

Use tools listed at http://metamanagement.comm.nsdlib.org/OAIvalidation.html or similar tools to ensure discoverability of your resources (http://www.openarchives.org/service/listproviders.html). A new repository should be officially registered http://www.openarchives.org/data/registerasprovider.html. This process allows assigning the repository an identifier which is unique in the OAI world. All oai-identifiers should then be derived from that repository identifier and therefore be authenticated in the OAI world.

7. Communicate to service providers any change that might affect them

If a repository is to be unavailable for an extended period, or your identifier scheme is going to change wholesale, consider informing service providers who routinely harvest your site.

For more information, please refer to

- the Digital Library Federation working group on best practices for OAI and shareable metadata **http://oai-best.comm.nsdl.org/cgi-bin/wiki.pl**

- the NSDL Metadata Primer
http://metamanagement.comm.nsdlib.org/outline.html
The NSDL Metadata Primer has a checklist of implementation details for an OAI repository at
http://metamanagement.comm.nsdlib.org/OAIServerChecklist.html.
It highlights implementation details to which you should pay particular attention.

Software with support for the OAI Protocol

A selective list of turnkey solutions

Several features are not mandatory to comply with the protocol but are useful in the NSDL context. When assessing a turnkey solution, you should consider whether and how it handles:

OAI Sets	For large repositories containing multiple collections, the harvester can choose to harvest one or multiple OAI sets instead of the full repository
Multiple metadata formats	This must be supported to expose additional metadata formats over and above Simple Dublin Core
Resumption token and records Count	These support the ability to split the metadata harvest into multiple chunks for better handling of large amount of content
Persistent deleted records	Allows you to indicate to the harvester when a record has been removed. This facilitates incremental harvesting and obviates need for frequent full harvests.
Granularity of datestamp	Repository updates can be recorded by second or by day. The more precise, the better. A harvester can perform a better incremental harvest based on more granular datestamp.

- **CWIS** : http://scout.wisc.edu/Projects/CWIS/
  Collection Workflow Integration System (CWIS), is a turnkey open source software package developed by the Internet Scout Project as part of the NSDL initiative. It includes a search engine, a recommender system, OAI and RSS servers, and more. ***Features*** : version 1.3.1 supports OAI sets, resumption tokens, oai_dc and nsdl_dc, and the OAI-SQ extension for searching via OAI-PMH.

- **ContentDM** : http://contentdm.com/
  It is a good content management system for displaying images and multimedia content. ContentDM allows pricing according to the collection size, making it affordable for small collections.
  Features : version 3.7 does not handle multiple metadata formats, handles deleted records, resumption tokens and sets. The system allows compound objects but does not allow you to define datestamp granularity level.

- **Digitool** : http://www.exlibrisgroup.com/digitool.htm
  Digitool is developed by ExLibris, especially adapted to library systems. It is coupled with METALib which allows harvesting OAI records. Digitool also allows sharing using Z39.50. ***Features***: handles multiple metadata formats, has a bug that currently prevents from harvesting sets larger than 1000 records. Handles resumption token (up to 1000 records) and sets but not deleted records.

- **DSpace** : http://www.dspace.org/
  Open source software developed by HP and MIT. It is intended to hold born-digital asset.
  Features: version 1.2 DSpace handles OAI sets, resumption token, deleted records.

- **EPrints** : http://software.eprints.org/
  The oldest OAI repository software is an open source solution developed at the University of Southampton, it is adapted to grey literature for ePrint repositories (self-archiving and open access)
  Features : version 2.3.0 default configuration only handles OAI-DC format

- **Encompass**: http://encompass.endinfosys.com/
  Created by Endeavor Information Systems, EnCompass is a digital library system for academic and research libraries, which includes a number of features such as Z39.50 target.
  Features: OAI coming.

- **DLXS**: http://www.dlxs.org/
  Created by the University of Michigan. The OAI feature can be used in conjunction with the collection manager portion of DLXS (collmgr) and the xpat search engine. It runs using a MySQL or CSV database.
  Features: Version 11a (of DLXS) : sets, resumptionToken, collection description, DLXS BibClass to oai_dc mapping

Packages can also be plugged to an existing system

- **OAICat**: http://www.oclc.org/research/software/oai/cat.htm
  Java servlet Web application created at OCLC.
  version 1.5.30

- **VTOAI Perl Data Provider**: http://www.dlib.vt.edu/projects/OAI/software/vtoai/vtoai.html
  OAI toolkit written in Perl at Virginia-tech.
  Version 3.05

- **XMLFile**: http://www.dlib.vt.edu/projects/OAi/software/xmlfile/xmlfile.html
  Created at Virginia Tech, it is a data provider module that operates over a set of XML files that contain the metadata.
  Version 2.1 –not fully robust, datestamp and character escaping issues, does not support deleted records

- **DLESE OAI software**: http://dlese.org/oai/index.jsp
  This software contains a data provider and a data harvester. Stores XML files.
  Version 3.05 does not seem to correctly support multiple metadata formats but easy to implement.

Other systems with at least some support for OAI: FEDORA (University of Virginia / Cornell) http://www.fedora.info/, Greenstone (Unesco / University of Waikato in New Zealand), ADLIB for museums, libraries, archives http://www.adlibsoft.com, Keystone http://www.indexdata.dk/keystone/ (IndexData). A current list of available packages is published by OAI at: http://www.openarchives.org/tools/tools.html